THE $2.00

JACG NEWSLETTER

JACG

THE JERSEY ATARI COMPUTER GROUP

VOLUME 8 NUMBER 8 BBS : 201-298-0161 OCTOBER 1988

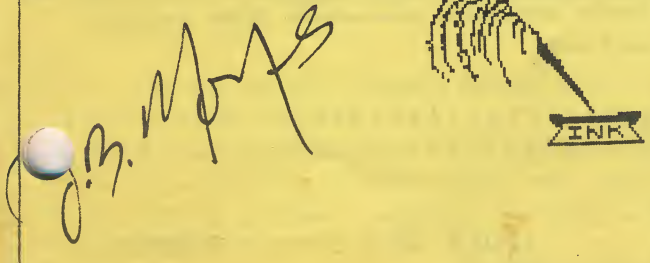
FROM THE EDITOR'S DESK

Autumn is here, and with it, a new year for the JACG...membership has stabilized, after having fallen in the past year, and hopefully new ST'ers abd 8-bitters (yes, 8-bitters [not bitter 8-bit owners!]) will see the value of the JACG, and join. Of course, if one is already a member...PLEASE renew your membership when due!

Nomination and election of officers is fast upon us...here is YOUR chance to have a "say" in the future of the JACG...as they say, "Vote early and vote often!" On the serious side, for those of you who attend our meetings and read this NEWSLETTER; what you don't see is the continuing dedication of the Executive Board (I exclude myself [Ah, so humble!])...who, besides their visable duties at the meetings...work "off-line" for the club...I hope you read the minutes of the Executive Board meetings. The Board members take an evening from their own personal schedules for the good of the JACG. They deserve a healthy round of applause for the past year of dedication!

Thanks are also due to the many members who have contributed time, NEWSLETTER articles, and demonstrations over the past year; and to those who by their attendance and membership have helped to ensure the existence of the JACG...thanks, folks!

'til next month....



IN THIS ISSUE ...

CALENDAR OF EVENTS

Nov. 12, 1988	JACG Monthly Meeting: Election of Officers
Dec. 10, 1988	JACG Monthly Meeting
Jan. 14, 1989	JACG Monthly Meeting: Children's Special

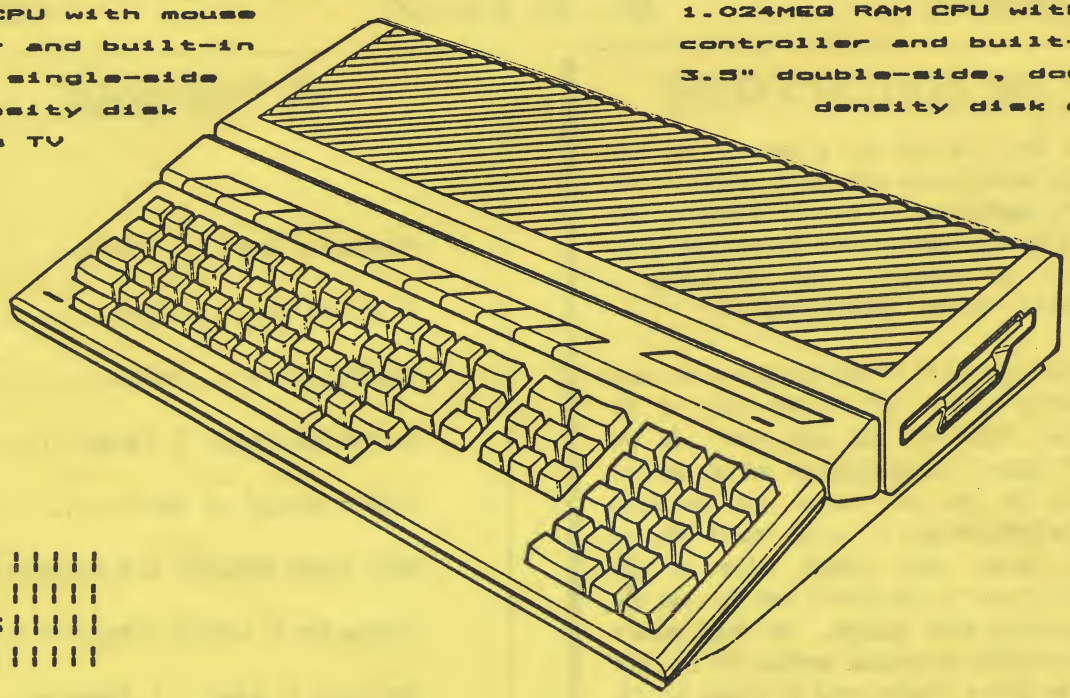
ATARI® ST Computer System

68000 MICROPROCESSOR – 8 MHZ CLOCK – RS232 SERIAL PORT – PARALLEL PORT
HARD DISK DRIVE PORT – CARTRIDGE PORT – SECOND FLOPPY DRIVE PORT – 192K ROM
MIDI INTERFACE – TOS OPERATING SYSTEM – GEM DESKTOP – 512 COLORS
MONOCHROME 640 X 400 RESOLUTION – COLOR 640 X 200 RESOLUTION

90-day Warranty – Over-the-Counter Exchange

520ST^{FM}.....$549.95
512K RAM CPU with mouse controller and built-in 360K-3.5" single-side double-density disk drive plus TV output

1040ST^{F}.....$709.95
1.024MEG RAM CPU with mouse controller and built-in 720K 3.5" double-side, double-density disk drive



MONO SYSTEMS

520ST^{FM} CPU with .....$ 699.95
ATARI SM124 Hi-Res B&W Monitor

1040ST^{F} CPU with.....$ 859.95
ATARI SM124 Hi-Res B&W Monitor

COLOR SYSTEMS

520ST^{FM} CPU with .....$ 789.95
MAGNAVOX CM8505 Med-Res RGB Color Monitor

1040ST^{F} CPU with.....$ 949.95
MAGNAVOX CM8505 Med-Res RGB Color Monitor

520ST^{FM} CPU with .....$ 889.95
ATARI SC1224 Med-Res RGB Color Monitor

1040ST^{F} CPU with.....$1049.95
ATARI SC1224 Med-Res RGB Color Monitor

GEMINI ENTERPRISES
86 Ridgedale Avenue
Cedar Knolls, NJ 07927

(201) 267-0988

PRESIDENT'S REPORT

by Linda Peckham

THE JACG NEEDS YOU!

The disturbing non-event took place at the September meeting. No one -- not one single person -- volunteered to do a demonstration for the October meeting! No eight-bit users, no sixteen-bit users. No one.

This club cannot operate without volunteers! Whether a volunteer gives one or two hours a year, or one or two hours a day, each and every volunteer is needed -- and the more, the better.

To be blunt, we need -- the Jersey Atari Computer Group needs -- more than the twenty-five dollars a year you pay for joining. We need YOU. We need your articles; your description of a favorite program -- or a disliked one, or how you use your computer, or what you think the computer club should be doing. We need your demonstrations at the club meetings -- whether it be the newest game, or an old favorite, an application program, utility program, commercial program, public-domain program -- whatever you use, and know. Yes, we even need your questions, especially as we try to expand the question and answer session before the meeting for both computers. We will have boxes in the back of the auditorium, for questions to be written down. These questions will be answered, if possible, during the q&a session, which will run from about 9:30 to 10:00. These questions and answers will later turn up in the newsletter. If, for some reason, you are unable to make the club meeting, you can still help in this area -- if you have a question, write it down and send it to the appropriate vice-president, and we will try to answer it in one of the subsequent newsletters.

I know that many people are reluctant to get up in front of a group and talk, but it is really not that hard. After all, we are your friends, those of us sitting out there. Here are a few things you should think about, in preparing for a demonstration:

* Know your program! You need to be able to answer questions about it, and be able to show the most important features of it.

* Length -- Most demonstrations should run about ten to twenty minutes in length.

* Practice -- If you can, run through your demonstration at home. Keep in mind that the projection screen is not that easy to read from the table's location. We do now have a small Black and white TV for the 8-bit users, but, at present, have nothing for the ST users.

* Jot down some notes -- Noting a few brief points on a piece of paper can help you remember -- it's easy to forget something, when you're in front of an audience. Don't write everything out -- few things are as boring as someone reading their demonstration. But a few words or phrases to remind you of the points you intend to make can help.

CONTINUED on page 14

NOISE FROM NOYES

by Dave Noyes

As you will have noticed - those of you who were at the October MEETING - there was a lot of ST software (and a little 8-bit), for sale in the lobby, courtesy of ATARI EXPLORER magazine. This was software given to the magazine, gratis, by the developers/producers, for review purposes. As ATARI EXPLORER felt that the sale for profit of such software was not appropriate, they established a nominal (primarily $10.00) price for the software. Further, all proceeds from the sale were slotted for a local charity...the Mendham Hills Chapel. Thank you 'EXPLORER'!

RUMOR time...for those of you acquainted with the U.K. ATARI publication, PAGE 6; it appears that, as had previously been the case with ANALOG, trouble looms on the horizon. Of course, we have ATARI EXPLORER, ANTIC, and ANALOG...but the demise and potential demise of magazines covering ATARI (for me, 8-bit machines) is a bit, to say the least, scarey. CREATIVE COMPUTING, SOFTSIDE, ROM, HI-RES, HOME COMPUTER MAGAZINE and coverage in FAMILY COMPUTING and COMPUTE!; just to mention those that immediately come to mind.

The point of it all (and although I specifically address the 8-bit - when summer arrives, can fall (or 'the fall') be far behind? The point of it is that without user (and we ARE the USERS) support (yeah, I know about corporate ATARI's support) the house will eventually come tumbling down. For instance, when was the last time you plunked down $$dollars$$ at a retailers for software for your ATARI (flea markets don't count). Do you have something in your software library that you shouldn't? Shame, NO support, NO software. Unless dollars go to retailers; they won't get to wholesalers, distributors, manufacturers, or developers.

My point, it's quite simple...if the bulk of the ATARI community supports the ATARI and ATARI-related business the way the bulk of the members support this organization (the JACG); there will not be an ATARI in the future. Just as a few devoted ATARIANS can not infuse the ATARI economy with enough $$dollars$$, a few hard-working, devoted members cannot keep the JACG forever.

Elections are upon us (JACG); please take the time to consider what both you and others can do for the JACG...and also consider what you and others may not have done for the JACG. We must pull together. The JACG is still a vital organization. Let's keep it so...or more!

'til next month...

ST Disk Library
JACG_LIB.000

By the time you read this, the JACG Catalog Disk, JACG_LIB.000 should be available. This disk will contain a disk catalog program, DC.TOS, and three data files -- JACGLIB1.CAT, JACGLIB2.CAT and JACGLIB3.CAT. This files will contain the file information for JACG Library Disks #1-50, #51-100 and #101-(150). READ_ME.1ST will have information on how to use the DC.TOS program, and the SORTER.TOS program which is also included on the disk. The description field in the catalog files includes such information as 1) the monitor the program runs on, 2) memory requirements 3)type of program, and a description of the program. Note that not all the files on the disk will be in the catalog -- Resource files (RSC), data files, and other supplementary files will not be listed, due to space and readability considerations. As room permits, there will also be text file versions of the three cat files, designed for use by the SORTER.TOS. This utility will be able to sort the text files by the information set up in the first columns of the description field.

Disk of the Month -- #116: Utility Disk #7

Startups and Ramdisks

This month's featured disk is a new utility disk, containing boot-up programs, and ramdisks. This disk is not set up to use automatically, so you will need to read the documentation, to create a boot-up disk. The programs included on the disk include:

SUPERBOOT 4.1 -- This is a newer, expanded version of the Superboot that is included on disk #66. (4.1 could not be put on #66 because of a lack of room.) This program allows a user to set date and time, display a picture, and select auto-boot programs, accessories, GDOS Assign.Sys files, and other files at bootup. The programs and accessories can be selected individually at each turn-on session, or the function keys can be set up to automatically determine the selected programs.

DESKMANAGER 2.1 -- Is another boot-up program that allows programs and accessories to be selected at startup.

RAMBABY -- A ram disk and print spooler accessory from Intersect that allows both to be turned on after boot-up, and be later turned off. The accessory will also write and information file, and will look for it on bootup, bringing the two up automatically.

MAXIDISK -- A bootup ram disk program from Germany, which can be configured to set the time, and copy files into the ramdisk.
HRAMDISK -- Another ram disk

FSELV60 -- A public domain replacement for the File Selector Box. It does not have all the features of the commercial Universal Selector, but it does allow changing from disk to disk without using the keyboard, and shows the memory size of the files.

September Error

It should be noted that the September column contained an error the Games Disk #13, Dungeons and Dragons, is Disk #115, not 113.

NEW DISKS

#117 -- Clip Art #6: Whimsey. This is another disk of .TNY files containing monochrome clip-art, for use in such programs as Publishing Parter and Publish ST. This disk is from the Current Notes Library, #288. The files were originally on the Macintosh, ported to the ST, and contain various (mostly small) images of humor, and animals. The disk also contains a slide program, and several conversion programs.

#118 -- EASYDRAW Utility Disk. Another Disk from Current Notes (CN #229) This disk includes three fonts in various sizes (CHICAGO, COURIER and CALIG), and eighteen .GEM pictures. These fonts should be usable by other GDOS programs.

#119 -- SHEET 1.7 This is a public-domain spreadsheet program. This version includes graphics. An accessory version is also included.

#120 -- Clip Art #7: Seasons. This clip art is shareware, downloaded from GEnie, scanned art, converted to TNY. The pictures include scenes from the four seasons of the year -- spring, summer, autumn and winter. TNYVIEW and TNYSTUFF are also inluded on the disk. (Disk is 10-sectored, 400K.)

#121D -- Casting D'Enterprise, by Propulse. This disk is a French import, running on color one-meg system, demonstrating the animation and graphics capabilities of the ST. It runs about 7.5 minutes. (From CN Library, disk #228).

STREPORT.988 -- This disk marks a beginning of a new set of disks for the library. STREPORTS is a weekly electronic magazine available from the major telecomm services (GEnie and Compuserve), as well as a number of BBSs. For the convenience of those who do not have modems, the library will make the magazines available on disk, starting with this month. The disks will be available for three months at the sales table. Later than that, they will be available at special request.

UPDATED DISKS

One number that makes its return to the official library list is #012 NEOCHROME. This disk now contains the last public-domain version of Neochrome, version 0.6. (The docs, are for 0.5, with some additions.) Also on the disk are pictures in the .NEO format, including some from the old disk #2, which were in the PIC format, a format virtually unused now. Even without all the features of the commercial version, this is a very nice little paint program, though for low resolution only. For those of you who may not have become acquainted with NEOCHROME, one of its nicests features (in my opinion), is the magnified section of the screen (the program sets up the lower half of the screen for a toolbox and the magnification).

CONTINUED on page 14

DESKTOP PUBLISHING

by Donald Forbes - JACG

Desktop publishing has now become a reality for many business word processing users since Version 5 of WordPerfect is now available for the business PC. They will be in a position to publish high-grade newsletters using their laser printers and current resources. They will also be able to generate professional looking documentation for all their software with a minimum amount of effort.

WordPerfect addressed our user group at the September meeting. Bill Hawes, sales representative for New York and New Jersey, gave us a slide presentation and donated a copy of the ST version as a door prize.

WordPerfect claims that they are outselling the competition two to one. They have brought out WordPerfect for the Atari ST computer based on Version 4.1 for the MS/DOS machines. I expect that once they establish a market for the Atari machines, they will come out with a desktop publishing version that will take advantage of the superb graphics of the Atari ST.

The ST is a big seller in Europe, both to the home and business markets, where buyers are not beholden to the Big Blue image of IBM. Atari has also repackaged their old 800 eight-bit machine, not as a computer, but as a GAME machine. It lists for about $200 and sells for about $150, with a detachable keyboard, and the option to attach a disk drive (and thereby turn it into a computer!). The dealers refused to handle the Atari 800 as a computer, but they are more than happy to market it as a game machine, complete with joystick and light gun. The product is a success (it sells itself with little or no advertising).

If I am not mistaken, someone at the last user group meeting said that Atari is now among the Fortune 400 corporations in terms of profitability. The European market is where the money is.

Desktop publishing traces its history back to 1455, the date of the printing of the earliest book printed from movable type, the Gutenberg Bible (some 40 copies exist today, including copies at the Huntington, Morgan, New York Public, Harvard and Yale university libraries).

Johannes Gensfleisch zur Laden zum Gutenberg was born on an unknown day between 1390 and 1400. He invented a method of printing from movable type that was used without important change until displaced by the computer. The unique elements of his invention consisted of (1) a mold with punch-stamped matrices (metal prisms used to mold the face of the type) with which type could be cast precisely and in large quantities, (2) a type-metal alloy, (3) a new press derived from those used in wine making None of these features existed in Chinese or Korean printing, or in the existing European technique of woodblock printing.

Gutenberg was a inventor who placed perfection above promptness, while his business partners wanted a safe and quick return on their investment. Gutenberg's dream was to create a way of mechanically reproducing medieval liturgical manuscripts without losing any of their color or beauty of design. They sued, and Gutenberg lost: he was ruined financially on November 6, 1455.

He died in poverty and almost blind in 1468. Were he alive today, he would be a millionaire many times over.

We have come a long way since Gutenberg. His Bible had no title page, no page numbers, and no innovations to distinguish it from the work of a manuscript copyist. But he brought the written word to the public, and thus is responsible for publishing as it has been known for five hundred years.

Until now, publishing has belonged to the few: those with the knowledge, the money, and the time to engage in a complex process involving many players. The process begins with an idea. It continues with a writer, who originates the material to be published. An editor modifies it for accuracy and editorial slant. The designer gives the publication its shape. A typesetter sets the words. An illustrator enhances the ideas with pictures. A paste-up artist assembles the publication. A commercial printer completes the process. Each one of these skilled steps takes days or weeks, and must be precisely coordinated. Just managing the process can be a full-time job.

Now we have come full cycle since Gutenberg. With desktop publishing using WordPerfect or any other package, you can now perform all those complex functions and produce quality printed material in much less time and at much less expense.

Desktop publishing today is the use of personal computers to create newsletters, flyers, brochures, advertisements, catalogs and manuals, and combines layout, design and integration of graphics to create what looks like a professionally published document. Ventura Publisher and PageMaker are the best known software packages. When you use them, however, you must use two separate programs: a word processing program to create the text, and then retrieve it into a desktop publishing program, both of them powerful but complex and difficult to master.

WordPerfect needs only one program. You can import drawings and illustrations from other graphics programs, and squeeze, expand, scale, rotate, move or invert them, and embed them in boxes, horizontal and vertical rules, and tables. You can add borders, captions, and gray shading. Leading, kerning (changing the space between wide and

skinny letters), and word and letter spacing are available. All this, without having to master a complex program. It is not, however, a WYSIWYG (what you see is what you get) program. You cannot just manipulate the final output on the screen, and therefore you have to plan everything carefully in advance.

Desktop publishing can be a successful hobby as well as a business. Bob Denton, one of our ST-owning members, publishes a newsletter on salt-water aquarium fish (an esoteric field about which most fresh water fish collectors know nothing).

The door prize, a copy of WordPerfect for the ST, was won by our longtime member Stefan Andrus Burr, once of Lockheed and AT&T who is now a professor of computer science at City College in New York City. Some years ago he organized a conference on the Mathematics of Networks for the American Mathematical Society. Last summer he organized a similar short course on chaos and fractals.

The centennial celebration of the American Mathematical Society in Providence RI in early August attracted 1700 people. The weekend before the meeting, the AMS sponsored an enormously successful short course on chaos and fractals. The course attracted a record crowd of about 500. The diverse crowd included many graduate students, participants from industry and laboratories, and even some high school teachers.

The audience heard presentations on such topics as the horseshoe map, chaotic attractors, Julia sets, and iterated function systems. In addition, they saw computer generated illustrations and films representing the mathematical objects explored in the course.

Now that Stefan has a copy of WordPerfect, perhaps he will give us a write-up for the newsletter of the highlights of his short course.

MESSAGE from the SYSOP

Gary Gorski - JACG

In order to stay current on this BBS, you must call at least once every 60 days. If you do not call, the sytem will automatically delete you. You then will have to **REREGISTER** to be a user on the JACG BBS... If you would like to check your EXPIRATION date, [Q]uit to main menu, then [V]iew options.

AN 8-BIT FLIGHT SIMULATOR II TOUR

WITH
SUBLOGIC SCENERY DISK #11

Dave Arlington - JACG

This article is a step by step tour of a flight from Buffalo, New York to Toronto, Ontario using an 8-bit Atari computer, Flight Simulator II, and the new SubLogic Scenery Disk #11 which covers Western New York and Southern Ontario as well as western Pennsylvania, northern Ohio, and most of the state of Michigan. This article is written especially for the Western New York Atari Users Group and the Toronto Atari Federation and has been submitted jointly to both newsletters.

If you have FSII and rarely use it due to your lack of flying ability, you will like this article since it gives step by step instructions and is Atari specific. You will be able to take off, fly and land by reading along. If you're fairly adept as FSII, you can discover some of the features and quirks of the SubLogic Scenery disks as well as seeing how they treat our area. And of course, I hope this article inspires everyone to rush out and buy Scenery Disk #11 so SubLogic will continue to support the 8-bit Ataris as they have done in the past.

If you already have Scenery Disk #11, you can follow along with this flight step by step. If not, read along anyway. When and if you acquire the disk, you can come back and fly along with the rest of us.

Load up Flight Simulator using either the disk version or the XE Game System cartridge. (Later on, I'll compare the two versions. I think you'll be suprised at which I prefer.) Insert the Scenery Disk #11B and press Control-E to log the disk in. Make sure you use disk 11B since that is the one with the Buffalo-Toronto area.
Press any key and then hit ESCape to enter the editor. The mode number should be zero. Now enter 110 for the mode number and you see the mode number changes to 10, the first of the user-defined modes.

Now, leaving everything unchanged EXCEPT the parameters I give you, type these parameters in:

NORTH: 17844
EAST: 19453
ALTITUDE: 725
HEADING: 215

When you are done, press Control-S to save this mode in memory. That way, if you happen to crash, you will be right back where you started. Hit ESCape to exit the Editor. The disk loads some information and there you are... Check your heading to make sure your compass (on the instrument panel, not the one up on the windshield) reads a heading of about 215-216. If so, fine. If not, hit Control-= to get things straight.

OK, where did I put you? If you hit 4 and then the Clear and Insert keys to adjust your radar, you will see your plane parked just off the end of Runway 23 of Greater Buffalo International Airport. Runway headings (or numbers) are compass headings with the last digit lopped off. So, Runway 23 has a heading of 230 degrees. If you were at the other end of this runway, you'd be pointing at Runway 5 with a heading of 50 degrees.

This is the major runway of Buffalo International, getting almost all the work as far as I can ever tell by driving by the airport almost every day. It is over a mile and a half long and stretches from where the New York State Thruway meets Transit Road up to where the Kensington Expressway crosses Dick Road. And here I had always thought the length of the runways were exaggerated so you could spot them better! So we have plenty of room to take off and as we do, we'll pass right over the Thruway Mall. (Use your imagination!)

Here's how to taxi for take off. Give one notch of throttle (the ↑ key) until the plane starts rolling and then take it off again (the + key). Keep using the radar view and steer a little to the right using the right aerilon key (the H key). As soon as you are pointed in the direction you like, hit the neutralize aerilon key (the G key) to straighten the plane out. Keep steering and straightening, using your radar to get perfectly lined up on the runway as best you can. When your heading is about 230, hit the brakes (spacebar) until you come to a stop. If you don't like how it comes out, hit Control-= to restart everything. Normally, airports don't appreciate little planes like ours stopping dead on the runway, but this is a special occasion.

Now for the actual takeoff. Switch to your out the windshield view. Press up elevator (the B key) once, wait a second and give it two quick presses. Put on one notch of flaps (the N key). Now put on full power, holding down the throttle key until you get 2450 RPMs (in the lower right corner). Watch your airspeed indicator (top row, all the way to the left). When the needle hits 80 knots, give one more press of up elevator. You lift off the ground. Wait about two seconds and take off that press of up elevator with a press of down elevator (the T key). When you are climbing 500 feet per minute (The vertical speed indicator or VSI is the third from left on the bottom row) take off your notch of flaps (the Y key). The VSI needle will rise up a bit, drop and then rise up again. When it gets to 1000 feet per minute, give one press of down elevator and two notches of down throttle for an RPM reading of 2250. Soon you will settle into a nice steady 500 foot per minute climb.

Whew! Now that that's taken care of, you can take a glance out your windshield to see how Buffalo looks from the air. Wow, there's nothing there. Just green. No city, no streets, no nothing, just green. I admit I was a little disappointed the first time I made this flight. I thought at least the New York State Thruway would make it into the simulator. We'll solve the mystery of the missing Buffalo later. For now, let's turn due west and head out over downtown Buffalo and see if anything is there. At about 1800 feet altitude (I hope you know which dial that is!) start a right turn to a heading of 270 degrees while climbing. Give two quick presses of right aerilon and wait until the wing on the turn and bank indicator (the instrument in the lower left corner) lines up with the dot. Then hit the G key to neutralize. Try to keep the wing lined up with that dot as you turn. First with a press of the H key and then center with the G key. At about 265 degrees, give two quick presses of the left aerilon key (the F key). Use the G key to neutralize just as the wings are coming level. You should flatten out within a degree or two of 270 and be headed approximately towards where the Niagara River dumps into Lake Erie. Hmm, still no signs of civilization.

When you get to about 2700 feet altitude, lower your power three more notches to 1950 RPM. You will level out into straight and level flight at about 2800 feet altitude. Take a glance out your right front or right window. There is a stick or something that looks like one of the buildings you see out the side window when parked at Meigs Field in Chicago. After studiously poring over my brother-in-law's real life FAA Sectional chart for this area, I have determined that these 'sticks' are what the FAA terms as obstructions. Tall buildings or power poles or smokestacks or just other things you would not want to fly a low flying plane into. I'm pretty sure that 'stick' out the right window is Erie County Medical Center as it is the most dominating building in that area. You will notice another 'stick' out the left front window. That one I have not identified.

When the Niagara River disappears under the bottom of your windshield, use the same instructions as above to make a gentle turn to the right again to a heading of about 335 degrees. That should point you right up the Niagara River and over Grand Island. There we see the first signs of the Thruway starting in Grand Island and crossing over into Niagara Falls, New York. Keep your nose pointed up over the middle of Grand Island until the north edge of the island passes off the bottom of the windshield. Then make a gentle left turn this time to a heading of 325 degrees. We're going to look at the Falls!

We're a little high to get a good look at the Falls and the Rainbow Bridge, so cut your power four notches to set up a 500 foot per minute descent. When you reach 1500 feet altitude, put the four notches of power back on. Now here is a word or two about flying in the Falls area. First, don't try to make any quick turns. The simulator is drawing so much scenery in this area, that your controls will come to an almost complete freeze while the computer tries to keep up. What will happen is that you will overcontrol and be hopelessly out of control. Secondly, this will admittedly not be the best view of the Falls since we are flying from the high area to the low. The best views will be to look out the left side and behind you as we pass. Don't forget to look out the right side as well to look up the Niagara Gorge as we go by. It really is a gorge. For a real good look at the Falls, see the notes at the end of this article. You might pass almost directly over the Rainbow Bridge. Look straight down for a neat view.

As we leave the Falls area, climb back to 2800 feet by adding three notches of power and taking them off again at 2700 feet. As you leave the Falls behind, make a gentle right turn to a heading of 0 degrees, so you are pointing due north into the middle of Lake Ontario. If you look out the left front window, you will see the airport at the lovely city of St. Catherines. Just to the west of that is what looks like a river that cuts all the way through from Lake Ontario to Lake Erie. That is the Welland Canal and is the main route for shipment of goods across the Great Lakes.

Time to try out the instruments. First hit the Pause key (the P key) while I discuss with you something mysterious about the sectional chart that comes with the disk and the Toronto area. Pull out the chart and look at it. It shows three airports in the Toronto area; Buttonville, Downsview, and Toronto Island? OK, the first two, I have no trouble with. But that airport labelled Toronto Island looks like Toronto International. Toronto Island really does exist but is indeed on an island right on the edge of Lake Ontario about halfway between where Downsview and well, that other airport are on the map.

So I pulled out the documentation and it lists only the three airports; Buttonville, Downsview, and Toronto Island. When I typed in the coordinates for Toronto Island, they gave me some trouble (which I'll explain later) but it definately was the Toronto Island airport and not Toronto International I was at. So did that mean that SubLogic really left Toronto International out of the simulation??? Hard to believe, but since there are no coordinates listed anywhere in the docs, there's only one way to find out. Fly there! So let's!

If you notice, there is a VOR tower right at that mislabeled airport with a frequency of 113.3. Hit P to unpause and tune it in on your OMNI bearing indicator. Use Control-N and the Clear and Insert keys to change the number before the decimal sign and use Control-N twice and the same keys to change the number after the decimal point. When you have that tuned in, your DME (Distance Measuring Equipment) kicks in and tells you how far you are from that tower. Now to tune one more thing. My brother-in-law's actual FAA chart shows a Runway 32 (heading 320 degrees) and a elevation at the airport (Toronto Intenational, if it's there) of 558 feet. So we'll use that as a guide.

Use Control-V and the Clear key to change the heading to 320 on top of the OMNI device. Keep flying straight north until the little white vertical line is almost at the center. When it is between the dot and the center, make a left turn to that heading of 320 degrees. We're going to follow this in and see if we find an airport.

While we look for signs of civilization in Toronto, I should point out that this is the first Scenery Disk issued by SubLogic that lets you fly from the United States and land in another country. The Western set of scenery disks would go up to the borders of Canada and Mexico but end there, so this disk marks sort of a historic first. I always love my trips to our neighbor to the north and this simulated trip is no exception.

Anyway, about 22 miles out we see our first sign of something on the shores of Lake Ontario, one of those obstruction thingies. More pop up and at about 18 miles out, a view out the right front window spots a patch of black that is the Toronto Island airport (the real one!). Gee, now we're getting closer and closer and Toronto looks as desolate as Buffalo did. I can sort of see that for Buffalo, but Toronto is Canada's New York City and should get something better.

Well, how do you like that? Just when I was about to give up on Toronto International being there and was going to turn around and head to Toronto Island, a disk access came up and all of Toronto sprung into view like I dropped out of a cloud. Beautiful! There is Toronto International up ahead, all the streets of downtown Toronto criss-crossing, and off to the right we see the CN Tower is fully digitized although it is not mentioned in the documentation.

All that's left is to get you on the ground. As soon as the disk accesses, drop power four notches to descend. When you reach 1600 feet, drop power another three notches and slowly give four notches of up elevator, seperating each notch of up elevator by about a second. Your height will vary over the next minute or so, but when it settles down, you should be level about 1500-1700 feet and at a speed of about 80 knots.

I hope Runway 32 is somewhere nearby, but if you are a novice at landing, DON'T, I repeat, DON'T worry about hitting the runway dead on your first time. You'll only crash and spoil the last 45 minutes. Landing ANYWHERE near the runway is good enough for today, unless, of course, you're a FSII pro, and then I want only the best.

As the end of the runway is about 1/3 of the way between the horizon and the bottom of your windshield, hit Control-I to add carburetor heat and put on a notch of flaps. Wait a second and give a notch of down throttle. When the edge of the runway touches the bottom of the windshield, give two quick notches of down elevator, follwed by two more notches of flaps. Now keep an eye on your altitude. When you reach 700 feet, give a notch of up elevator and then again at 600 feet. One more at 560 feet followed by cut of all throttle and you should hear the squeal of tires on the runway (or ground, as it may be). Now hold the spacebar until you stop and congratulate yourself for a successful flight.

Now since SubLogic doesn't provide any information on Toronto International, here is the straight goods for making your own flights from there. NORTH: 18123, EAST: 19132, ALT: 558. There are 8 runways. 4 Left/22 Right, 4 Right/22 Left, 14/32, and 9/27. There is no refueling facility, nor is there an ATIS or an ILS approach. I don't have the dimensions of the runways, but they are all fairly large, as befitting a major airport.

OTHER INTERESTING TIDBITS

These are other notes of interest that intrepid newsletter editors may leave in or hold back for another month.

CREDIT WHERE CREDIT IS DUE: Some people may notice a similarity bewteen this article and many of the Flght Simulator books written by Charles Gulick including FLGHT SIMULATOR CO-PILOT, RUNWAY USA, 40 GREAT FLGHT SIMULATOR ADVENTURES, and 40 MORE GREAT ADVENTURES et al. I owe something to this author. I used to be one of those folks whose FSII program sat on the shelf where I would admire the box and wish I could fly it. After finishing FLGHT SIMULATOR CO-PILOT, I found my wishes had come true and now I fly around wherever I like. I reccomend unequivocably that anyone wishing to learn how to fly the simulator should buy this book and the other three as well. End of plug.

If you do use any of Mr. Gulick's books to learn to fly, you should add 1024 to any elevator setting to any mode that starts off

the ground. The elevator setting for operational neutral for the Atari 8-bit is 37887, not 36863 as it is in Mr. Gulick's Commodore 64.

THE MYSTERY OF THE MISSING BUFFALO: Get your plane back in Buffalo again, either retyping the parameters at the start of the article or hitting Control-=. Now enter the editor and change the time to 19:10. Exit the editor and go into radar view. This is Buffalo at dusk, and lo and behold, there is the City of Good Neighbors in all it's glory. So here is the trick; Buffalo and a few other cities show up in this new Scenery Disk only at twilight or nighttime. They are invisible in the daytime.

I think the reason is in the Scenery Disks for the 16 bit versions (ST and Amiga). There Buffalo shows up as a yellow color. But since the 8-bit versions (including the C-64 and Apple versions) are limited to only 4 colors, it leads to some wierd coloring decisions on the part of SubLogic; like coloring the cities the same as the ground. I prefer the handling of cities in the intial Scenery Disks where the cities showed up as black patches of pavement.

RADAR SOMETIMES LIES: Type in the parameters given for Toronto Island. It doesn't matter what heading you use. Now quickly look out all your windows and back out the front again. Wait a second... SPLASH! Looks like we're sitting right in the middle of some water. Where's the ground? When the simulator resets, quickly go into radar view. Zooming to any level shows your plane safely on ground and you can stay there indefinately without splashing. Once you go to look out the windshield though, SPLASH!, into the drink you go.

Now use these parameters: NORTH: 18112, EAST: 19205, HEADING: 60, ALTITUDE: 0. Press Control-= to get it right. Take off straight ahead and look out the back window as you take off. There the mystery is revealed. Although radar says you were on solid ground, Toronto Island is really an island and the SubLogic parameters put you in the water just offshore.

By the way, while I was here, I used SLEW controls to put my plane right next to the base of the CN Tower and then used altitude slew to move up the side of the CN Tower like an elevator. My intention was to measure the height of the CN Tower. The tip of my wing came level with the top at a little over 2000 feet above sea level. Checking the FAA Sectional, it lists the CN Tower as being 2055 feet above sea level. Now that's accurate!!

A GOOD LOOK AT NIAGARA FALLS: Type in these parameters and sit back and just watch: NORTH: 17896, EAST: 19336, ALTITUDE: 1000, PITCH: 357, HEADING: 200, AIRSPEED: 117, THROTTLE: 11223, ELEVATORS: 37887. When you're ready, hit Control-= to watch it again.

One of the times as you go by, look out the right front. You will notice what looks like a little canal of blue there. Switch to your right side and right rear views as it goes by and follow it. Looks pretty wierd, doesn't it? Not like any canal of water I've ever seen.

Try this. Go into the Editor and put in a layer of clouds with tops at 3000 and bottoms at 2000. Now exit and try the flight again. It turns out that thing is really the Skylon Tower in Niagara Falls, Canada. It just happens to be colored the same blue as the sky so you can't see it very well. Remember what I said about SubLogic's wierd coloring decisions before? In the 16 bit versions, the Skylon Tower is in red, but of course, in the 8-bit version, there is no red color and so we landmarks that are the same color as the sky. Again, this same fluke appears in the Commodore and Apple versions as well.

DISK VS. CARTRIDGE: Like many people I was anxious to try the new XE Game System FSII cartridge and see the difference between the two. My disk version says it is version 1.05 and the cartridge is version 1.07. They are essentially the same with these few differences.

The main advantage to the cart is the instant loading time and no copy protection that doesn't like your disk drive. In addition, all the New York-Boston, Seattle, Chicago, and Los Angeles scenery areas are contained right in the 128K cart and are instantly loaded when needed. The drawback to this is without the disk access to clue you in, you might miss a neat scenery feature if you are not looking out the right window at the time. Another difference with the cart is all the instruments on the panel with digital readouts are in red, while the disk version readouts are in blue. This is simply a one-pixel shift of the artifacted colors used.

There is one serious (to me, anyway) flaw in the cartridge version that has prompted me to go back to the disk version. Whenever you call a control tower using your COM radio in the cartridge version, the temperature is always 75 degrees (even in New York in Winter), the wind is always 330 degrees and the landing runway is always 31, whether or not a runway 31 exists at the airport you are calling. So while the name of the airport changes, the report itself is always the same. Since I love calling the towers whenever I can, I went back to using the disk version where the information is accurate.

Well, that is all to this article. I hope it is not too long to print and that it inspires some people to get into flying like I have these last few months. Good luck and see you over Lake Ontario!

Programming the ST
with
Personal Pascal
Part 2

Paul Machiaverna - JACG

Last month I showed a couple of very simple TOS programs for you to try typing in, compiling, and running with OSS Personal Pascal (PP). This month I will cover writing some simple GEM programs. The simplest GEM function to use is the Alert Box. Alert boxes are what you see when a program informs you to do something, or something is wrong. You can see an Alert box on you screen by taking the disk out of your disk drive and opening the drive icon (double-click on the drive icon). The ST will try to read a disk in the drive a few times then display an Alert box telling you that an error has occurred. This is the type of GEM function I will cover here this month.

One comment I must make before we begin is concerning remarks placed in Pascal programs. In last month's article it was not clear as to what was a remark and what was the actual program. Unfortunately, the required left and right curly brace characters used to tell Pascal that the lines are remarks did not appear because the editor of this news letter is using the 8-bit Atari computer which does not support these characters as text. So, from now on I will use the older notation which is (*, to mark the beginning of a remark, and *), to mark the end of a remark. For example, (* This is a Pascal remark *). However, I suggest that you replace all (* with the left curly brace, and *) with the right curly brace at all times when you copy the programs. In some cases the right and left curly braces are used to give instructions to the compiler. In this case the (* *) pair will not work.

Let's start by covering a few important points about writing GEM programs. PP has to be told that we are writing a program to be compiled for GEM. This will make the compiler use the proper routines for creating a GEM program, and the linker will use the proper libraries for producing an executable file with the PRG file extender. So, the first thing to do is pull down the options menu from the PP manager and select compile and link for GEM. Next we have to include a library file in our source code to inform the compiler where to find the routines to support the GEM calls in our program. Finally, our program has to ask the ST if it can use the GEM environment when we try to run our executable file. The reason for this is that a GEM application needs to be initialized, meaning that our program will be using the GEM functions. Now, let's consider our first, simple GEM program.

As I stated before, Alert boxes are the simplest of all GEM functions to use. However, their simplicity of use also includes their simplicity in what they can do. Alert boxes are useful for simple halting of a program to inform the user of a error, to instruct them to do something, or to get simple predetermined input. Alert boxes can contain an icon to enhance their appearance and convey the nature of it's reason for being on the screen. You can choose one of three icons or none at all. The three possible icons are the exclamation point, the question mark, and the stop sign. No icon in an alert box is good for program titles or conveying simple information to the user. The exclamation point is good for grabbing the users attention to important information, like if you wanted to say 'Clicking on OKAY will format your disk!' That's an example of a operation which would require the user to make a very important choice. The question mark is good for telling the user that some unknown problem has occurred or that the program does not understand some input. The stop sign is good for instructing the user to stop and doing something like 'Be sure that your printer is online.' The impact of icons on the user are tremendous. Pictures convey information much quicker and vivid to the user than is possible with words. For this reason, it is a good idea for you to think about how you want an Alert box to appear.

Take a careful look at the program below which will show all the possible Alert boxes. Incidentally, only one Alert box can be displayed on the screen at once. Now type in the program being very careful to replace all (* *) shown with the respective curly braces. Where you see (*$I GEMSUBS.PAS*) in the program you must replace the (* *) or the program will not compile. After you compile and run the program to see it's action we will discuss each line in detail to understand what it is doing.

```
(* Demonstration of the four types of the GEM Alert Boxes.  They
are:
  1) No Icon displayed
  2) Exclamation point in an inverted triangle
  3) Stop Sign
  4) Question mark in an inverted triangle

     Written by Paul Machiaverna - JACG    September 12, 1988  *)

     Program Four_Alert_Boxes ;

     (*$I GEMSUBS.PAS*)  (* Include the GEM routines *)

     Var
Choice : Short_Integer ;  (* value returned by alert box *)

     Begin
If Init_Gem >= 0 then  (* request the use of GEM *)
Begin
Choice := Do_Alert('[0][¦   Alert Box¦ with No Icon  ¦][  Next
]',1) ;
Choice := Do_Alert('[1][¦  Alert Box¦  with an¦ Exclamation Point
¦][ Next  ]',1) ;
Choice := Do_Alert('[2][¦ Alert Box¦  with a¦ Question MArk¦][  Next
]',1) ;
Choice := Do_Alert('[3][¦  Alert Box  ¦ with a¦ Stop  Sign¦][  End
]',1) ;
Exit_Gem  (* we are done using the GEM routines *)
End
End.
```

Congratulations! You have just completed a GEM program. If you have problems with compiling and linking the program the first thing to check is that you used the curly braces instead of the (* *) on the line which includes the GEM routines, as discussed above. If you still have problems, make sure that you chose compile and link for GEM from the PP manager pull down menu. Now let's take the source code apart and understand each line.

The program starts off with a comment which tells what the program does. I can not emphasize the importance of commenting your source code enough. Spend a little time including comments when you write a program instead of spending a lot of time later trying to figure out what your program does. The next line is the standard required Program statement found in every Pascal program. The most important line of this program is the ($$I GEMSUBS.PAS$) directive. It tells the compiler to include the GEM subroutines required to write a GEM program. The source code, or any other GEM program, will not compile with out this directive. GEMSUBS.PAS is a file which is written in Pascal syntax and includes all the calls necessary for the linker to properly make an executable GEM program. The file is included on the PP disks. The integer value called 'Choice' is used to hold the value returned by the Alert box. This value is the choice of the user and is made by clicking on a box with mouse or hitting return on the keyboard to choose the default choice.

Now we have to request the use of GEM by using the Init_Gem function. From how I understand it, GEM can only handle a certain amount of work at a time. You have to ask if GEM can initialize an application before you use it, such as displaying an Alert box. The value returned must be greater than or equal to 0 if we are to run our GEM application. A negative value basically means that GEM is telling us that our application cannot be initialized for use. The successful return value (>= 0) of Init_Gem is only useful when writing Desk Accessories. Thus, this return value is not needed for any other use in our program.

The actual call to the Alert box to make it appear on the screen is done with the Do_Alert function. Do_Alert() is called a function because it is called by assigning it's return value to a variable. Therefore, we cannot call up the Alert box by simply writing Do_Alert(). We must write it as Choice := Do_Alert(). Choice was declared as a Short_Integer because this is the type of value returned by calling the function. The () parentheses denote that there are parameters required by the function. These parameters tell GEM how the Alert box is to appear and how to handle the button(s) of the box. The buttons of an Alert box are the rectangles to which the user moves the mouse pointer and clicks on to exit the box and return a value to the program. The parameters of the Alert box are 1) a string which contains information about the type of icon to display, the text to appear in the box, and the names of the buttons contained in the box, 2) a number which tells GEM which button to consider the default, if any. The default condition of a button is when the user can invoke the highlighted button by either clicking on it or hitting the return key. You may choose to have any button from 1 to 3 as the default or no default by coding a 0 for this part of the string. Note that each part of the string in contained by left and right brackets, [and]. The vertical lines in the text part of the string denotes a new line for each vertical line shown. This means that you can have more than one line of text in an Alert box.

The program continues by displaying each type of Alert box possible. It will show one box, wait for the user to click on a button or hit return, and show the next type, until all four have been shown. The last function we encounter in this program is Exit_Gem. This tells the computer that we are done using GEM. You should not call this function if the call to Init_Gem returned an unsuccessful initialization value of less than zero. Unpredictable results can occur if this rule is violated.

The use of Alert boxes are simple, but can be frustrating. They are simple because you only have to write very little code to create and display them (one line!). They are frustrating because you don't have a lot of control over their appearance and there are some cautions to abide by to avoid a system crash. The PP owner's manual gives you a list of cautions when programs which use Alert boxes. As for what to consider, I'll point out a few things in the use of Alert boxes. The only control you have over the size of the Alert box is dictated by the string which you enter as the parameter to the function. Getting text to be centered in the box requires some experimenting at first. No patterns can be used to enhance the appearance of the box. Alert boxes are always centered on the screen. Your program will stop dead waiting for the user to click on one of the buttons or hit return if enabled with the default condition parameter. Overall, remember that Alert boxes are intended to grab the user's attention (hence, 'Alert') to something important when running a program.

Experiment with creating all different types of Alert boxes by writing simple programs to test them. Learn how to properly center the text in an Alert box. Know how to use the right type of icon to convey the nature of the appearance of the Alert box. Alert boxes are ideal for learning how to write programs for GEM. Eventhough there are limitations to there use, Alert boxes are an integral part of many GEM programs and are very useful because of their attention grabbing quality.

I hope you found this discussion of Alert boxes interesting. Above all, you now know some basic considerations for writing GEM programs. The GEM user interface is very powerful. When writing any GEM program you should think in terms of the user and do whatever is necessary to make your program easy to use. Next month I will break from GEM programming and discuss how to write TOS programs which tap the potential of the screen formatting routines available in PP which allow for easy formatting of text which are not available in standard Pascal. Finally, I will make the source codes of the programs shown in these articles available on the JACS BBS for those who don't like to type in programs. However, I will leave the compiling up to you for the experience. See you next month.

A Joy(stick) to Behold

J.S.Roborecky - JACG

When I received my Wico Ergostick for evaluation I thought here is a copy of a similar joystick that is now on the market.How wrong I was!

When I removed the Ergostick from its box I knew I had a joystick to behold.

the first thing you notice is its soft gray color and non-slip surface.The Ergostic also has an extra long cord for those who like to sit back from their TV or monitor.

When you hold the Ergostic in your hand you notice that it is narrower than brand X and fits more comfortably.The positive movement of the stick as it turns the microswitchs on and off makes moving the players,at an angle,on the screen a lot easier.Playing your favorite action game for a long time the Ergostick will not start to slip out of your hand like brand X does.The gray dimpled soft plastic surface prevents slippage even after long usage.

On a sour note I must chastise WICO for putting such a small trigger button on the Ergostick.The position and deep indentation of the trigger button makes pressing it tiresome after long periods,as the trigger finger falls in an unnatural angle on it.

I Think WICO has come up with a product of rugged construction that will last a long time and is worthy of consideration.

DAISY DOT II DIED AGAIN

or

SAM CORY WHO DID IT

S. Cory - JACG

In case you tried to run DDII. Suprise! Your Disk Librarian did it again. I have finally figured it out. The computer is out to get me. It is so strong Bob Mulhearn has coined the phrase "The Cory Factor". I will not go into the gory details of what and why DDII did not work.

I appologize to all you buyers of DDII and Oh yes, the Ol'Hackers disk #2. There was nothing on that disk. I forgot to check both disks before they were sent to the meeting.

So here we go again. Bring in yours for replacement. If any of these fail. Do not worry I will not go away. The only way to get rid of Sam Cory, Disk Librarian--Shoot him. A lot of words to say I am VERY VERY sorry for the inconveniences you have endured.

J.A.C.G. EXECUTIVE MEETING

R. MULHEARN, SECRETARY - JACG

The meeting was called to order at 8:00 PM by president Linda Peckham with Linda, Jack Rutt, John Dean, Dave Noyes, Gary Gorski and Bob Mulhearn present, constituting a quorum.

Linda opened with a discussion of the upcoming elections. All elected positions of President, Vice-President 8 bit, Vice-President 16 bit, Secretary, and Treasurer are open for nominations. Jack Rutt and John Dean said that they would be running again. Linda brought up the problem of lack of demos for the October meeting and some alternatives were discussed. Linda finished with a report on where she was with the 16 bit 000 catalog disk which will sell for $2.00 or $1.00 with a returned disk to be updated quarterly.

John Dean followed with the problems he that he has with the 8 bit Disk of The Month. Jack reported that the state of the treasury is remaining stable. Bob followed with a discription of the secretary's duties. He also discussed the changing of the clubs mailing address with the possible renting of a P.O. box. It was decided to leave any final decission with the new board to be elected in NOV. Dave brought up the subject of a multi-color newsletter for December and will check on costs, as well as changing the cover color of our monthly newsletter. He also brought up an offer to sell software at greatly reduced prices at the swap meet with all proceeds going to charity. The offer was met with approval.

Gary ended the meeting with a discussion of the BBS, its condition and some minor problems he is having. The board gave him a free hand to solve those as he chooses. He also brought up the idea of a children's special meeting in JAN.89.

Programming in ATARI BASIC

J. Beebe

Multiple statements may be hung on the end of an IF-THEN, separated by colons. The code following the THEN will be executed when the IF-THEN statement tests TRUE. All code past the THEN will be executed up to the permissible maximum line length of three screen lines. The code following the THEN is never executed if the statement tests False. For example:

```
10 FOR X = 1 TO 25
20 IF X>=6 AND X<=10 THEN ? X
30 NEXT X
40 END
```

This program prints:

```
 6
 7
 8
 9
 10
```

The equal sign tests for EXACTLY equal to, and 5.9999 is NOT equal to 6. This can be tricky. It is common practice to use < and > to test for numbers, or <= and >= when possible, to avoid numeric errors resulting from multiplication or division causing slight inequality. Strings may be tested, and < and > will test for alphabetical order of the first letters of strings.

Again, don't trust your eyes and brain to know whether you've arranged your AND's and OR's, parentheses and lo[illegible]correctly. It always LOOKS right. Use correct and incorrect input, and make sure by trial and error that everything is correct.

Some BASIC's support an ELSE addition to the IF-THEN statement, but Atari BASIC does not. Don't worry, we don't really need it. You can always arrange code to achieve the effect you want. When an IF-THEN statement is False, execution resumes at the very next line, and the IF-THEN is NOT executed.

The opportunities, and combinations are [illegible]ess, and knowing a few good "hacks" is what separates new from experienced programmers. We will continue to learn about IF-THEN statements for a long time.

COUNTERS:

Often, we wish to keep track of how many times we have gone through a loop. We may want to do something every other trip through, or count when to end, or keep track of how many times we've done a thing. This is done with COUNTERS, which are +1 or [illegible]ng statements placed within the body of the loop. These let the computer do the counting, freeing us from this chore, which is the whole idea behind using and programming computers.

These often look like:

```
10 X = X + 1

70 TOTAL = TOTAL - 5

25 COUNT = COUNT + 1
```

N[illegible] the way these are coded. C[illegible]rs say, "a variable equals itself plus something". The variable name appears on each side of the equal sign, with the "plus something" on the right. Remember a LET statement takes what's on the right of the equal sign, and assigns it to what's on the left of the equal sign. It's a little hard to think so elementary at first, X = X + 1, but becomes second nature soon. I'm serious, they're hard to catch on to coding.

SUMMERS:

SUMMER is not a proper programming term, but that's what I call these. Summer's are utilities much like counters, but rather than counting, they total or add together or accumulate a running sum total number. For example, a summer in a payroll program keeps track of how much we have earned to date in the year.

These are coded much like counters, but often have a variable for a "plus something" on the right hand side. Some examples:

```
15 TOTHOURS = TOTHOURS + HOURS

10 YTDSAL = YTDSAL + WEEKSAL

55 SUM = SUM + (14.7%TAX-INTEREST)

10 TOTAL = TOTAL + ADDON
```

If counters usually tell us "how many" a thing is now, then summers keep track of "how much" we have now. Trust me! This will make sense someday. Sometimes when a program you are writing is acting up, and you can't figure it out, you can install some counters and/or summers in the code, and RUN it. Break in with the BREAK key, and ask it to print the value of a counter or summer in immediate mode at the READY prompt, like "? SUMM" or "? TOTAL" or "? COUNT". It may be helpful to see if it's looped more or less times than you had thought, or perhaps you find it got stuck after a set number of loops. When the program's debugged, take out the lines.

Putting It All Together

Let's write a program that puts this all together, that you can continue to work on and develop, and learn on. Let's imagine you are a teacher, who needs to average student grades for the entire semester, and assign a letter grade. If doing this by hand, we would add all of a student's grades, divide by the number of grades, and find a numerical average. This would then be compared to a range of numerical grades and letter grades. We must count the number of grades each student has, to find the correct number to divide by. This if often confusing when students have a different number of grades in a class, and could lead to an error.

Let's write a program that calculates these grades on our Atari computer in BASIC. When calculating the average, it must keep track of how many grades we enter for each student, automatically choosing the correct divisor. The program should prompt you for correct input, and error check it for correctness. It should stop taking grades and display the average when we type in a grade of 999 to signal the end of input.

We may add to this program at some point, so we'll start numbering it at 1000. Leaves lots of room "below."

```
1000 REM **  EZ-GRADE  **
1010 REM JACKSON B   6/87
1020 REM
1030 REM CLEAR,INITIALIZE,PROMPT,IN
1040 ? CHR$(125):TOTAL = 0:COUNT = 0
1050 ? :? :? "Enter grade ";
1060 INPUT GRADE
1070 REM
1080 REM TEST,INCREMENT,SUM,LOOP
1090 IF GRADE = 999 THEN 1160
1100 IF GRADE <0 OR GRADE > 100 THEN
GOTO 1050
1110 TOTAL = TOTAL + GRADE
1120 COUNT = COUNT + 1
1130 GOTO 1050
1140 REM
1150 REM AVERAGE AND OUTPUT
1160 AVG = TOTAL / COUNT
1170 ? :? "The AVERAGE for these ";
COUNT;" grades is ";AVG
1180 ? :? "(Press 9 to enter more
grades)  ";
1190 INPUT M
1200 IF M = 9 THEN 1040
1210 END
```

This program uses a counter (COUNT) to keep track of the number of grades entered, and a summer (TOTAL) to keep a running total. Entering a 9 after an average, will continue the program. Any other key will end the program.

Note that the counter and summer are initialized (set to zero) OUTSIDE the main loop.

Sample Problems:

PROBLEM 4

Add lines to EZ-GRADE to determine a letter grade A, B, C, etc, in additional to a numerical average. (HINT - add a pile of IF-THEN's after line 1170 that test whether input is between 70 and 79, 80 and 89 etc, and assign a letter to a variable. Use a one character string variable to hold the letter grade, that is dimensioned outside the loop (line 1135.)

PROBLEM 4A

Modify EZ-GRADE so that instead of typing in 999 to signal the end of data, you simply press RETURN without having entered a grade (a NULL entry.) Have it prompt whether you really want to stop entering data before it calculates the average and letter grade. (HINT - you need a string variable to bring in a plain carriage return. The VAL function will translate your string input into

a number.)

The way you test for an empty carriage return (RETURN) is to test the input for equality with two sets of quotation marks back-to-back with nothing between them as:

```
195 IF NAME$ = "" THEN 255
```

This tests True for a null input, and goes to line 255. You MUST hit the RETURN key to pass an INPUT statement in BASIC.

PROBLEM 4B

Write a GUESS THE SECRET NUMBER program. Use the following line early in the program, which will generate a random number between 1 and 100, and assign it to the variable SNUM. Write your program to prompt the player for a number from 1 to 100, and give feedback if the guess is higher, lower, or equal to the secret number (a winner.) Write it to loop 10 times, then signal that the 10 guesses are up, and you lose!

```
35 SNUM = INT(RND(0)*100+1)
```

PROBLEM 4C

Write a program that asks you to guess the capitol of 5 states, and gives you a grade from 0 to 100 on your answers.

This concludes Lesson 4 of Learning to Program in Atari BASIC. Be sure to catch Lesson 5 which includes:

```
FUNCTIONS
  INT(), RND(), VAL(), SQR(),
  ASC(), CHR$(), ABS(), SIN()
More tricks and treats
```

Let's back up for one moment, and recall a rap we had in Lesson 1, in which I identified 3 tasks beginners face in learning to program.

1. Learning to operate the hardware.

2. Learning to program.

3. Learning the BASIC language.

Once you learn most of the basic programming concepts and structures, you will never need to learn them over again, if you pursue other languages. For instance, you will not relearn the concepts of testing, branching and counters. You will simply ask "What's the syntax to...test, loop, branch, etc." It is natural to be very confused right now. Do not quit because you are confused. That simply means your brain is making the correct thinking adjustments. It is natural to dream about programs and problems when working on new projects. Relax, it's not permanent, and some of your best ideas just pop into your head after days of muddled thoughts.

Contact me at:

Jackson Beebe
Prairie Data Fields
807 W. Hill Street
Urbana, IL 61801

or CompuServe 72550,317

CONTINUED from Page 3

Disk Prices:

JACG_LIB.000 (catalog) $2.00 ($1.00 with old JACG_LIB.000 disk)

Members:

Disk of the Month $3.00

Regular Disk $4.00

Non-Members:

$6.00

Mail-order, add $1.00 per disk.

Send order to:

Bill Garmany, Jr.,
13 Wellington,
Livingston,NJ
07039

CONTINUED from page 4

* ST Users only: Make sure your program runs on a color monitor! Also, the system set up normally has only single-sided drives. Contact me if you need double-sided! I have a disk drive I can bring -- but you must let me know ahead of time.

* Talk to the audience, not to the screen. This is difficult, given our current setup. But try to spend at least part of your time facing the audience. Also, remember that we have not been using microphones. Try to speak loudly enough so that everyone can here.

* Finally, don't be nervous! If that's your first time up there, remember, everyone goes through a first time. We're rooting for you, to show us something interesting about a topic we all like well enough to spend Saturday mornings attending a meeting. We're on your side.

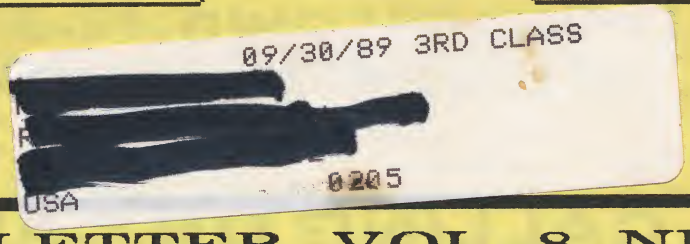
J A C G

JERSEY ATARI COMPUTER GROUP

8 CRESCENT ROAD

PINE BROOK, NEW JERSEY 07058

BULK RATE
U.S. POSTAGE
PAID
PINE BROOK, NJ
PERMIT # 56



JACG NEWSLETTER VOL. 8, NUM. 8

OCTOBER 1988

EXECUTIVE COMMITTEE

PRESIDENT Linda Peckham
111 Paterson Avenue, Totowa, NJ 07502, 201-790-3061

VICE-PRESIDENT ST Linda Peckham
111 Paterson Avenue, Totowa, NJ 07502, 201-790-3061

VICE-PRESIDENT 8-BIT John H. Dean
RFD #2 Box 788, Sussex, NJ 07461, 201-827-3902

SECRETARY & MEMBERSHIP Robert P. Mulhearn
8 Crescent Road, Pinebrook, NJ 07058, 201-575-0067

TREASURER Jack Rutt
52 Dacotah Avenue, Rockaway, NJ 07866, 201-625-0723

EDITOR David B. Noyes
3 Ann Road, Long Valley, NJ 07853, 201-852-3165

LIBRARIAN Sam Cory
P.O. Box 7, Towaco, NJ 07082, 201-334-4443

ADVERTISING {open}

SALES Gary J. Gorski
313 Sheridan Avenue, Roselle, NJ 07203, 201-241-4554

PRESIDENT EMERITUS Doug Van Hook
40 Meadow Lane, Clifton, NJ 07012, 201-472-0637

MAIL ORDER LIBRARIAN Bill Garmany, Jr.
13 Wellington, Livingston, NJ 07039

ASSISTANT LIBRARIANS
[8-BIT] Doug Van Hook, Dave Green, Bill Garmany, Jr.
[16-BIT] Linda Peckham, Charlie Miller, Eric Jacoves

BBS SYSOPS
Gary Gorski, Doug Van Hook, Paul Machiaverna, Tom Shoosmith

J.A.C.G. MEMBERSHIP APPLICATION

DUES: U.S. 3rd Class Mailing, Canada, Mexico $25.00
U.S. 1st Class Mailing, Foreign subscriptions 31.00

____ Renew ____ New ____ Former
____ 8-bit ____ 16-bit

NAME_____

Address_____

City_____

State/Country/Zipcode_____

Home Phone Number_____

Date_____ 1st Class____ 3rd Class____

MAIL TO: Robert Mulhearn
8 Crescent Rd, Pinebrook, NJ 07058

Call our Bulletin Board
at (201)-298-0161